Title	量子アルゴリズムによる近似文字列出現頻度問い合わせ (計算機科学基礎理論とその応用)
Author(s)	小林, 健了; 小野, 廣隆; 定兼, 邦彦; 山下, 雅史
Citation	数理解析研究所講究録 (2005), 1426: 225-231
Issue Date	2005-04
URL	http://hdl.handle.net/2433/47286
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

数理解析研究所講究録 1426 巻 2005 年 225-231

量子アルゴリズムによる近似文字列出現頻度問い合わせ

小林 健了 (Takenori Kobayashi) * 小野 廣隆 (Hirotaka Ono) †
定兼 邦彦 (Kunihiko Sadakane) † 山下 雅史 (Masafumi Yamashita) †

概 要

本研究では文字列出現頻度問い合わせに対する量子アルゴリズムを考察する．これは，入力としてテキスト T とパターン P が与えられたとき，T 中に P が出現する回数を求める問題である．文字列探索の量子アルゴリズムでは，時間計算量が $\tilde{O}(\sqrt{n}+\sqrt{m})$ であることが過去の研究により示されている．ここで，n, m はそれぞれ T, P の長さである．対して，古典アルゴリズムによる時間計算量は，KMP 法により $O(n+m)$ である．

上記問題の解決にあたり，入力として、ある小さい確率以下で誤りを含む 0/1 のデータ列が与えられたとき，データ列中の 1 の数を求める問題について考察する．この問題に対して通常の数え上げをそのまま適用できないため，majority voting により誤りを高確率で吸収することで，量子数え上げを文字列出現頻度問い合わせで適用できるようにした．また，周期的なパターンが与えられたとき，配列の総和計算を求める量子アルゴリズムを設計した．これらを用いて時間計算量の考察を行う．結果として $\tilde{O}(\sqrt{m}+\sqrt{nt})$ (t: 問題の解) 時間量子アルゴリズムを得た．

また，ミスマッチ許容文字列探索及び数え上げに対する量子時間計算量を紹介する．

1 はじめに

量子アルゴリズムは現在使用されている古典アルゴリズムと比較して，一部の問題に対してより高速な演算を行う計算機構として注目を集めている．それらの代表的な問題として整数因数分解問題 [6] と探索問題 [3] が挙げられる．[6] では整数因数分解に対する多項式時間量子アルゴリズムが発見された．一方，この問題が古典アルゴリズムにより多項式時間で解けるかどうかは知られていない．[3] では探索問題に対する $\Theta\left(\sqrt{N/t}\right)$ 時間アルゴリズム (以下，量子探索と呼ぶ．) が発見された．ここで，N は入力データ長，t は探索すべき解の数 (未知) である．これによって [6] で見られるほど劇的な計算時間の減少は起きていないが，時間計算量の下限が発見されていることが興味深い．

量子探索の関連研究として，量子数え上げ [1] と文字列探索に対する量子アルゴリズム [5] が挙げられる．[1] では前者に対する $O\left(\sqrt{Nt}\right)$ 時間アルゴリズム (正確な数え上げ) および $O\left(\epsilon^{-1}\sqrt{N/t}\right)$ 時間アルゴリズム (近似数え上げ) を与えている．また，[5] では後者に対する $\tilde{O}(\sqrt{n}+\sqrt{m})$ 時間アルゴリズムを与えている．ここで，n は入力テキスト長，m はパターン長である．

本研究では文字列出現頻度問い合わせに対する量子アルゴリズムを与え，その時間計算量を解析する．また，問題解決の際に必要となる，エラーが起こり得るデータに対する数え上げと配列の総和計算に対して量子アルゴリズムの提案と時間計算量の解析を行う．結果として $\tilde{O}(\sqrt{m}+\sqrt{nt})$ 時間で動作する量子アルゴリズムを得た．また，ミスマッチ許容文字列探索および数え上げに対する時間計算量を紹介する．

2 準備

本節では量子アルゴリズムに関する基礎的な事項を概説する．

*九州大学大学院システム情報科学府, Department of Computer Science and Communication Engineering, Graduate School of Information Science and Electrical Engineering, Kyushu University
†九州大学大学院システム情報科学研究院, Department of Computer Science and Communication Engineering, Graduate School of Information Science and Electrical Engineering, Kyushu University

2.1 量子アルゴリズム

本節では量子アルゴリズムの基礎的概念である状態，重ね合わせ，および状態に対する変換であるユニタリ変換について概説する．

量子状態 $|\Psi\rangle$ は，P 個の基底状態 $|0\rangle,\ldots,|P-1\rangle$ が与えられたとき以下のように記述される．

$$|\Psi\rangle=\sum_{i=0}^{P-1}\alpha_i|i\rangle$$

これを状態の重ね合わせと呼ぶ．ただし α_i は $|i\rangle$ に対する振幅で，$\sum_{i=0}^{P-1}|\alpha_i|^2=1$ を満たす．状態 $|\Psi\rangle$ を観測すると確率 $|\alpha_i|^2$ で $|i\rangle$ が得られ，元の重ね合わせ状態には戻らない．

$|\Psi\rangle$ への演算はユニタリ変換により行われる．変換 U がユニタリであるとは，転置共役変換 $U^\dagger$ に対して $U^\dagger=U^{-1}$ が成り立つことをいう．入力として量子状態が与えられたとき，ユニタリ変換は量子状態を出力する．

2.2 量子探索・量子数え上げ

本研究では，問題の解決にあたり量子探索 [3]，量子数え上げ [1] およびそれらを改良したアルゴリズムを用いる．

量子探索は，長さ N のデータ f が与えられたとき，条件を満たしている任意のデータのインデックスを 1 つ返すアルゴリズムである．量子探索の時間計算量は $\Theta\left(\sqrt{N/t}\right)$ である．ここで，t は条件を満たしているデータの数で，未知であるとする．また，時間計算量は f に対する問い合わせ回数として評価する．このアルゴリズムでは条件を満たすインデックスの振幅を増幅するユニタリ変換である，Grover 反復 $\mathbf{G}_f$ を用いている．これは，W を Hadamarr 変換とするとき，

$$(\mathbf{G}_f)^jW|0\rangle\rightarrow k_j\sum_{x:f(x)=1}|x\rangle+l_j\sum_{x:f(x)=0}|x\rangle$$

を計算する．ただし，

$$k_j=\sin((2j+1)\theta),l_j=\cos((2j+1)\theta)$$

であり，θ は $\sin^2\theta=t/N$ かつ $0\le\theta\le\pi/2$ によって決まる．ここで $j\approx\frac{\pi}{4}\sqrt{\frac{N}{t}}$ をとると，高い確率で条件を満たすインデックスを観測することが出来る．

次に，$\mathbf{G}_f$ の周期性および t による周期の違いに着目し，t を求めることを考える．このアルゴリズムを量子数え上げと呼ぶ．周期の計算には量子 Fourier 変換を用いる．ここに，量子数え上げを行うアルゴリズム $\mathbf{Count}(f,P)$ を示す．

アルゴリズム 1 ($\mathbf{Count}(f,P)$)

1. $|\Psi_0\rangle\leftarrow W\otimes W|0\rangle|0\rangle$
2. $|\Psi_1\rangle\leftarrow\mathbf{C}_f|\Psi_0\rangle$
3. $|\Psi_2\rangle\leftarrow\mathbf{F}_P\otimes\mathbf{I}|\Psi_1\rangle$
4. $\tilde{f}\leftarrow|\Psi_2\rangle$ の第 1 レジスタの観測値
   ($\tilde{f}>P/2$ ならば $\tilde{f}\leftarrow P-\tilde{f}$)
5. $N\sin^2\frac{\tilde{f}\pi}{P}$ を出力

ここで $\mathbf{C}_f$ を

$$|m\rangle\otimes|\Psi\rangle\rightarrow|m\rangle\otimes(\mathbf{G}_f)^m|\Psi\rangle$$

と定義し，$\mathbf{F}_P$ を P 個の状態に対する量子 Fourier 変換，すなわち

$$|k\rangle\rightarrow\frac{1}{\sqrt{P}}\sum_{l=0}^{P-1}\exp\left[\frac{2\pi\imath kl}{P}\right]|l\rangle$$

で定義する．なお，$\imath=\sqrt{-1}$ とする．

アルゴリズム $\mathbf{Count}(f,P)$ の出力する解 $\tilde{t}$ は $8/\pi^2$ 以上の確率で

$$|t-\tilde{t}|<\frac{2\pi}{P}\sqrt{Nt}+\frac{\pi^2}{P^2}N$$

を満たすことが示されている．[1] では $\mathbf{Count}(f,P)$ を利用して正確な数え上げおよび近似数え上げに対する量子アルゴリズムを実現している．本研究では「数え上げ」としてそれらを用いる．ここで，近似的な数え上げとは，誤差パラメータ ϵ に対して $|t-\tilde{t}|<\epsilon t$ を満たす $\tilde{t}$ を求めるものとする．

3 文字列出現頻度問い合わせの定義

本節では，本研究で扱う問題を定義する．文字列出現頻度問い合わせは以下のように定義される．

問題 1 (文字列出現頻度問い合わせ)
入力：アルファベット Σ 上のテキスト T，パターン P．長さはそれぞれ n,m とする．
出力：T 中に P が出現する回数

T	a	a	b	a	b	a	c	b	a	c
P	a	b	a	b	a					
		a	b	a	b	a				
			a	b	a	b	a			
				a	b	a	b	a		
					a	b	a	b	a	
						a	b	a	b	a
	0	1	0	0	0	0	0	0	0	0

図 1: 文字列出現頻度問い合わせの問題例

問題の例を図 1 に与える．この場合の解は 1 である．古典アルゴリズムではこの問題が $O(n+m)$ 時間で解けることが知られている．本研究では量子アルゴリズムによる時間計算量が，正確な数え上げでは $\tilde{O}(\sqrt{m}+\sqrt{nt})$，近似数え上げでは $\tilde{O}(\sqrt{m}+\epsilon^{-1}\sqrt{n})$ であることを示す．

また，ミスマッチ許容文字列探索・数え上げは以下のように定義される．

問題 2 (ミスマッチ許容文字列探索・数え上げ)
入力：アルファベット Σ 上のテキスト T，パターン P，およびパラメータ k．長さはそれぞれ n, m とし，$k \leq m$ とする．
出力：T 中の，P との文字の一致している場所が k 箇所以上存在する長さ m の部分文字列の任意のインデックス *(探索)*，条件を満たす部分文字列の数 *(数え上げ)*

本文では $T[i]$ を文字列 T の i 番目の文字，$T[i..j]$ を T の i 文字目から j 文字目までの部分文字列とする．

4 エラーを考慮したデータの数え上げ

第 3 節で定義した問題を解く際，エラーを考慮したデータの数え上げを行う必要が生じる．本節では，その問題を定式化し，アルゴリズムを与える．

問題 3 (エラーを考慮したデータの数え上げ)
入力：*0/1* のデータ列 f'．各値の計算はオラクル $F(i)$ を通じて行われる．各 $f'(i)$ は真値を表し，$F(i)$ が真値を出力する確率は 3/4 以上とする．
出力：$t = |\{i|f'(i) = 1\}|$

この問題を [1] にある通常の数え上げで行う場合，$F(i)$ に対する問い合わせにより計算過程でエラーを生じてしまい，正しい解が得られるかどうかは自明ではない．この問題を解決するために，majority voting を用いた数え上げ $\mathbf{CountMaj}(f, P)$ を提案する．

アルゴリズム 2 ($\mathbf{CountMaj}(f', P)$)

1. $|\Psi'_0\rangle \leftarrow W \otimes W|0\rangle|0\rangle$
2. $|\Psi'_1\rangle \leftarrow \mathbf{C}'_{f',P}|\Psi'_0\rangle$
3. $|\Psi'_2\rangle \leftarrow \mathbf{F}_P \otimes \mathbf{I}|\Psi'_1\rangle$
4. $\tilde{f} \leftarrow |\Psi'_2\rangle$ の第 1 レジスタの観測値
   ($\tilde{f} > P/2$ ならば $\tilde{f} \leftarrow P - \tilde{f}$)
5. $N \sin^2 \frac{\tilde{f}\pi}{P}$ を出力

ここで $\mathbf{C}'_{f',P}$ を

$$|m\rangle \otimes |\Psi\rangle \rightarrow |m\rangle \otimes (\mathbf{G}'_{f',P})^m|\Psi\rangle$$

と定義する．また，$\mathbf{G}'_{f',P}$ を，ユニタリ変換 $\mathbf{G}_{f'}$ を $O(\log P)$ 回計算し，それらの majority voting により得られる値を出力する変換とする．このとき，次のことが言える．

補題 1 f を常に正しいデータを返す *0/1* のデータ列，f' を $\Pr[F(i) = f'(i)] \geq 3/4$ を満たす *0/1* のデータ列とする．また，$1 \leq i \leq N$ に対して $f(i) = f'(i)$ とする．このときアルゴリズム *1, 2* において，ある定数 c 以上の確率で $|\Psi_2\rangle = |\Psi'_2\rangle$ となる．

証明 majority voting によって，$\mathbf{G}_{f'}$ の反復適用を行ってもエラー率が定数で抑えられるとき，投票 1 回あたり有権者が $O(\log P)$ 人いれば十分であることを示す．

$p = \Pr[F(i) = f'(i)]$ とおく．このとき，仮定より $p \geq 3/4$ が成り立つ．今，1 回あたり k 人の有権者により majority voting を行うとする．このとき，j 人目の有権者の票に対応する確率変数 $X_{i,j}$ を次のように定義する．

$$X_{i,j} = \begin{cases} 1 & \text{if } F(i) = f'(i) \\ 0 & \text{otherwise} \end{cases}$$

また，i 番目の投票結果に対応する確率変数 X_i を

$$X_i = \sum_{j=1}^{k} X_{i,j}$$

として定義する．Majority voting の結果を F_M とするとき，$F_M(i) = f'(i) \iff X_i \geq k/2$ と定義する．

以上の情報から，majority voting の失敗率を Chernoff Bound[4] により見積もる．これより，1 回の majority voting が成功する確率

$$\Pr[F_M(i) = f'(i)] > 1 - \exp\left[-\frac{k}{24}\right]$$

が導かれる．

次に，ユニタリ変換 $\mathbf{G}'_{f',P}$ を m 回繰り返したときの成功率を見積もる．すなわち，f' が m 回とも $F_M(i) = f'(i)$ となる確率は以下のようになる．

$$(\Pr[F_M(i) = f'(i)])^m > 1 - m \cdot \exp\left[-\frac{k}{24}\right]$$

最後に，アルゴリズム 2 のステップ 2 が $0 \leq m \leq P-1$ において全て成り立っている必要がある．その確率を見積もる．

$$\begin{aligned} & \prod_{m=0}^{P-1} (\Pr[F_M(i) = f'(i)]) \\ > \quad & 1 - P^2 \exp\left[-\frac{k}{24}\right]. \end{aligned} \tag{1}$$

式 (1) の値を下から定数で抑えたい．今，ある定数 c で抑えらるための k の条件を求める．

$$1 - P^2 \exp\left[-\frac{k}{24}\right] > c$$

であることに注意すると，

$$k > 48 \ln P - 24 \ln(1-c)$$

が得られる．以上により証明された． ■

また，ステップ 2，4 の成功確率の独立性およびステップ 4 の成功確率が $8/\pi^2$ 以上であることから次の系が得られる．

系 1 アルゴリズム **CountMaj** の出力値 $\tilde{t}$ は $3/4$ 以上の確率で

$$|t - \tilde{t}| < \frac{2\pi}{P}\sqrt{Nt} + \frac{\pi^2}{P^2}N$$

を満たす．また，時間計算量は $O(P \log P)$ である．

ここで得た **CountMaj** を用いて，アルゴリズム **CountRM** および **CountEM** を与え，時間計算量を評価する．なお，前者は近似数え上げを行う量子アルゴリズム，後者は正確な数え上げを行う量子アルゴリズムである．また，**CountRM** の **Maj**(N, A) は，アルゴリズム A を N 回実行し，その majority result を出力する．

アルゴリズム 3 (CountRM(f', ϵ)**)**

1. $P \leftarrow 2$
2. $P \leftarrow 2P$
3. $\tilde{f} \leftarrow$ **Maj**$(\Omega(\log\log N),$ **CountMaj**$(f', P))$
4. $\tilde{f} \leq 1$ ならば 2 へ
5. **CountMaj**$(f', \epsilon^{-1}P)$ を出力

アルゴリズム 4 (CountEM(f')**)**

1. **CountMaj**$(f', \sqrt{N})$ を c 回計算 $\rightarrow$ $\tilde{t}$ を出力
2. **CountMaj**$(f', 20\sqrt{\tilde{t}N})$ を出力

補題 2 アルゴリズム 3 はエラーを考慮したデータの近似数え上げを行い，時間計算量は $\Theta\left(\epsilon^{-1}\sqrt{N/t}\log\epsilon^{-1}\sqrt{N/t}\right)$ である．また，3/4 以上の確率で $|t - \tilde{t}| < \epsilon t$ を満たす $\tilde{t}$ を出力する．

補題 3 アルゴリズム 4 はエラーを考慮したデータの正確な数え上げを $(3/4 - O(2^{-c}))$ 以上の確率で行い，時間計算量は $\Theta\left(\left(c\sqrt{N} + \sqrt{Nt}\right)\log\sqrt{N}\right)$ である．

証明 アルゴリズム **CountRM**(f', ϵ) においてステップ 4 の条件が満たされるとき，$P > 2\sqrt{N/t}$ となる [1]．ステップ 5 では **CountMaj** の第 2 引数 P は $P > 2\epsilon^{-1}\sqrt{N/t}$ を満たすので，**CountRM** の時間計算量は $\Theta\left(\epsilon^{-1}\sqrt{N/t}\log\epsilon^{-1}\sqrt{N/t}\right)$ となる．

次に，**CountEM** の時間計算量を解析する．ステップ 1 で $|t - \tilde{t}| < 2\pi\sqrt{t} + \pi^2$ なる $\tilde{t}$ が得られたとき，時間計算量は

$$\begin{aligned} \Theta(P\log P) &= \Theta\left(\sqrt{Nt}\log\sqrt{Nt}\right) \\ &= \Theta\left(\sqrt{Nt}\log\sqrt{N}\right) \end{aligned}$$

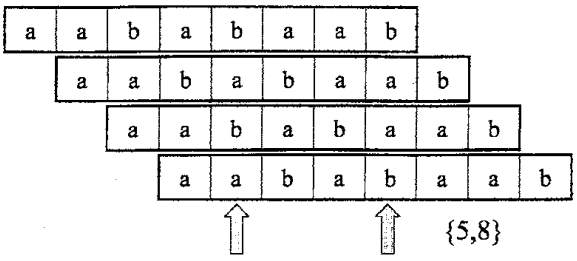

図 2: deterministic sample の例

が得られる．最後の等式は，$t \leq N$ による．これにステップ 1 における時間計算量 $\Theta\left(c\sqrt{N}\log\sqrt{N}\right)$ を加えると補題が得られる．なお，**CountRM**，が出力する値の評価については [1] による． ■

5 アルゴリズムと時間計算量

本節では，第 3 節で定義した問題 1, 2 に対する量子アルゴリズムを提案し，時間計算量を解析する．

5.1 問題 1(非周期的なパターンに対する量子アルゴリズム)

本節では deterministic sample を用いて，パターンが非周期的な場合の，文字列出現頻度問い合わせに対する量子アルゴリズムを提案する．パターン P が周期的であるとは，ある文字列 v，v の接頭辞 u，および 2 以上の自然数 k が存在して $P = v^k u$ と表されることをいう．

最初に deterministic sample を定義する．Deterministic sample の例を図 2 に示す．

定理 1 (Deterministic Sample 定理 [5]) P を非周期的な文字列とする．P を左から右へ 1 文字ずつずらして得られる $m/2$ 個のインスタンスを考える．これらのインスタンスに対して左から順に 1 から $m/2$ までの番号を振る．すると，以下の性質を満たすインスタンス f と *deterministic sample* と呼ばれる p の高々 $O(\log m)$ 箇所の場所の集合が存在する．すなわちインスタンス f が *deterministic sample* の全ての点でテキストと一致していれば，他の P のどのインスタンスもテキストと一致しない．

このアルゴリズムは deterministic sample を求める部分と数え上げを行う部分とに大別される．前者には [5] の $O\left(\sqrt{m}\log^2 m\right)$ 時間量子アルゴリズムを用い，本研究では後者の部分について言及する．

まず，テキストを長さ $m/2$ のブロックに分割する．パターンの存在判定や数え上げはブロック単位で行う．その上で次のオラクルを定義する．確率オラクル $h(i)$ は，i 番目のブロックにテキストと一致するインスタンスの数を返す．ここでは非周期的なパターンのみを扱っているので，各 $h(i)$ の値は 0 か 1 しか取らない．よって，このオラクルを判定オラクルとして扱う．オラクル $k(i,j)$ は，i 番目のブロックのインスタンス j が deterministic sample の全ての場所でテキストと一致しているとき，そしてそのときのみ 1 を返す．k の計算時間が $O(\log m)$ であることは明らかである．オラクル k と以下に定義するオラクル g を用いて確率オラクル h を構成する．ただし

$$g(i,j) = \begin{cases} 1 & \text{if } T[i+j-1] \neq P[j] \\ 0 & \text{otherwise} \end{cases}$$

である．

まず，$k(i,j)$ に対して量子探索を行い，deterministic sample でテキストと一致しているインスタンス j を 3/4 以上の確率で発見する．このステップの時間計算量は $O\left(\sqrt{m}\log m\right)$ である．次に，j が deterministic sample 以外の部分でもテキストに一致しているかどうかを調べる．これはオラクル $g(i',j') = 1$ を満たす j' の存在判定によって実現でき，そのような j' が存在しないとき，インスタンス j はテキストと一致する．このステップの時間計算量は $O\left(\sqrt{m}\right)$ である．

以上の操作によりブロック i にテキストと一致するインスタンスの存在が分かったとき，$h(i) = 1$ とし，そうでないときには $h(i) = 0$ とする [5]．確率オラクル h の時間計算量は

$$O\left(\sqrt{m}\log m + \sqrt{m}\right) = O\left(\sqrt{m}\log m\right)$$

である．$h(i)$ は確率オラクルであり，$1 \leq i \leq \lceil 2n/m \rceil$ であることに注意すると，h に対する近似数え上げおよび正確な数え上げに要する時間計算量はそれぞれ $O\left(\epsilon^{-1}\sqrt{n/m}\log\epsilon^{-1}\sqrt{n/m}\right)$，$O\left(\sqrt{nt/m}\log\sqrt{n/m}\right)$ となる．以上のことから，パターンが非周期的な場合，

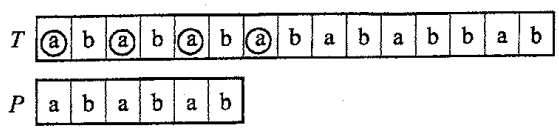

図 3: 周期的なパターンの例

2	1	1	0	0

図 4: 図 3 における各ブロック内のテキストと一致するインスタンスの数

文字列出現頻度問い合わせの時間計算量はそれぞれ $O\left(\sqrt{m}\log^2 m + \epsilon^{-1}\sqrt{n}\,(\log m)\log \epsilon^{-1}\sqrt{n/m}\right)$, $O\left(\sqrt{m}\log^2 m + \sqrt{nt}\,(\log m)\log\sqrt{n/m}\right)$ となる.

5.2 問題 1(周期的なパターンに対する量子アルゴリズム)

次に，本節ではパターンが周期的な場合の量子アルゴリズムについて検討する．まず，この場合では 5.1 節のアルゴリズムがうまく動作しない理由を述べ，次にアルゴリズムの提案とその解析を行う．

与えられたパターンが周期的である場合，非周期的な場合と比較して以下の 2 点の変化が起こる．

1. 各長さ $m/2$ のブロック内に，そのブロックから始まって，テキストと合致するインスタンスが複数存在する．

2. deterministic sample の計算が途中で失敗する．

最初の点については，定理 1 では，パターンが周期的であるとき各ブロック内のテキストと一致するインスタンスが高々1 個であることを保証していない．図 3, 4 に，周期的なパターンが与えられたとき，長さ 3 のブロックの中にテキストと一致するインスタンスの数が 2 個以上になる例を示す．また，$P = v^k u$ に対する deterministic sample の計算においても，あるインスタンス f とそれを $|v|$ ずつずらした全てのインスタンス以外が消された時点で，それ以上インスタンスを消すことができなくなる．

そこで，本節では，以下の方針で量子アルゴリズムを設計する．

1. 上記のインスタンスだけが残るまで deterministic sample を求め，これを用いて各ブロックにおけるテキストと一致するインスタンスの数を計算する．

2. 全てのブロックにおいて一致の数の総和を求める．

本稿ではステップ 1 には [5] を用いる．よって，ステップ 2 に対応するアルゴリズムを提案する．

この問題では，各ブロックに存在する，テキストと一致するインスタンスの数は高々 $m/2$ 個であることに注目する．ここで，次の問題を定義する．

問題 4 (上界つき総和計算) 入力 : 長さ N の配列 f，ただし各 $0 \le i \le N-1$ に対して $f(i) \le M = 2^l$ を満たす
出力 : $\sum_{i=0}^{N-1} f(i)$

この問題に対し，以下の量子アルゴリズムを与える．ただし，論理関数 $f_j(i)$ を，$f(i)$ の j 桁目を表すものとする．

アルゴリズム 5 (誤差 ϵ 以内の解を出力)

1. 全ての $0 \le j \le l$ に対して $\mathbf{CountRel}(f_j, \epsilon)$ を $O(\log N \log\log M)$ 回計算し，t_j をその majority result とする，

2. 加算の量子回路により $\sum_{j=0}^{l} 2^j t_j$ を出力

アルゴリズム 6 (正しい解を出力)

1. 全ての $0 \le j \le l$ に対して $\mathbf{CountEx}(f_j)$ を $O(\log\log M)$ 回計算し，t_j をその majority result とする

2. 加算の量子回路により $\sum_{j=0}^{l} 2^j t_j$ を出力

近似数え上げ，正確な数え上げの時間計算量はそれぞれ $O\left(\epsilon^{-1}\sqrt{N/t}\right)$, $O\left(\sqrt{Nt}\right)$ である．また，量子回路を使用する部分では入力 T, P に対する問い合わせは発生しない．ただし，この段階では量子アルゴリズムの出力を用いて計算を行うため，単純に

数え上げ結果を足すだけでは出力にエラーが蓄積される．よって，最初の段階での数え上げで majority voting を用い，出力の結果が定数以上となるようにする．以上のことから，上界つき総和計算の時間計算量はそれぞれ $O\left(\epsilon^{-1}\sqrt{N/t}\log N\log M\log\log M\right)$，$O\left(\sqrt{Nt}\log M\log\log M\right)$ となる．また，入力データにエラーが確率 1/4 以下で発生する場合，**CountRel** や **CountEx** の代わりに **CountRM** や **CountEM** を用いることで対応でき，全体の時間計算量もそれに応じたものとなる．

以上のことから，パターンが非周期的である場合，文字列出現頻度問い合わせの時間計算量はそれぞれ

$$O\ (\sqrt{m}\log^2 m + \epsilon^{-1}\sqrt{n}(\log m)^2\log\epsilon^{-1}\sqrt{n}\log n\log\log m),$$
$$O\ (\sqrt{m}\log^2 m + \sqrt{nt}(\log m)^2\log\sqrt{n}\log\log m)$$

となる．

5.3 問題 2 の時間計算量

本節では，第 3 節で定義した，問題 2 に対する時間計算量を紹介する．量子アルゴリズムでは，1 段階目において部分文字列とパターンの間の文字の一致の数の数え上げと，それらと k の比較を行う．2 段階目に，得られたリストに対して探索や数え上げを行う．2 段階目で使用するリストにはエラーが含まれるため，数え上げの際には majority voting を利用する．

ミスマッチ許容文字列探索・数え上げの時間計算量は，探索，近似数え上げ，そして正確な数え上げはそれぞれ $O\left(\sqrt{n}\sqrt{mt'}\log\sqrt{n}\right)$，$O\left(\epsilon^{-1}\sqrt{n}\sqrt{mt'}\log\epsilon^{-1}\sqrt{n}\right)$，$O\left(\sqrt{nt}\sqrt{mt'}\log\sqrt{n}\right)$ となる．

6 まとめと今後の課題

本研究では文字列出現頻度問い合わせに対する量子アルゴリズムを提案した．これらは，対応する古典アルゴリズムより早く動作する．

今後の課題として，majority voting を使わずに第 4 節に挙げた数え上げを行う手法について考えたい．エラーを考慮した探索の場合，[5] において majority voting により時間計算量 $O\left(\sqrt{N/t}\log\sqrt{N/t}\right)$ で計算可能であることが知られている．対して，[2] では同じ問題に対する $O\left(\sqrt{N/t}\right)$ アルゴリズムを与えている．量子数え上げについても探索の場合と同じような計算時間の短縮が可能であると考えている．

参考文献

[1] G. Brassard and P. Høyer and A. Tapp. Quantum counting. In *25th International Colloquium on Automata, Languages and Programming*, pp. 820–831, 1998.

[2] P. Høyer and M. Mosca and R. de Wolf. Quantum search on bounded-error inputs. In *Proceedings of ICALP 03*, pp. 291–299, 4 2003.

[3] L. Grover. A Fast Quantum Mechanical Algorithm for Database Search. In *28th ACM Symposium on Theory of Computing*, pp. 212–219, 1996.

[4] R. Motwani and P. Raghavan. *Randomized Algorithms*, chapter 4. Cambridge University Press, 1995.

[5] H. Ramesh and V. Vinay. String Matching in $\tilde{O}(\sqrt{n}+\sqrt{m})$ Quantum Time. *Journal of Discrete Algorithms*, Vol. 1, No. 1, pp. 103–110, February 2003.

[6] P. W. Shor. Algorithms for Quantum Computation : Discrete Logarithms and Factoring. In *Foundations of Computer Science*, pp. 124–134, 1994.